

## AMENDMENT TO CLAIMS

Please amend claims 1-13, 23, and 33 and add new claims 35-42, all as shown below. All pending claims are reproduced below, including those that remain unchanged.

1. (Currently Amended) A computer-implemented system to process an XML document, comprising:

a streaming parser operable to parse an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

a matching component ~~operable~~ to perform the steps of:

accepting an event from the stream of events from the streaming parser at one time;

keeping in memory only ~~a subset~~ said event of the stream of events at any said time;

performing a match on ~~an event in the subset~~ said event of the stream of events; and

notifying an observer ~~if when~~ the event is a matched event, wherein when the event is not a matched event the observer is not notified;

said observer operable to listen for the matched event and passing it to a user object; and said user object operable to handle the matched event.

2. (Currently Amended) The computer-implemented system according to claim 1, wherein: the XML document is represented in a hierarchical structure.

3. (Currently Amended) The computer-implemented system according to claim 2, wherein: the hierarchical structure is a tree with each node containing a portion of the document.

4. (Currently Amended) The computer-implemented system according to claim 3, wherein: the streaming parser is ~~operable to~~ generates the stream of events by:

traversing the XML tree and adding visited nodes into a data structure;

processing the nodes in the data structure and generating an event for each node;  
and  
appending the event to the output stream.

5. (Currently Amended) The computer-implemented system according to claim 4, wherein:  
the tree is traversed using a breath-first or depth-first search.
6. (Currently Amended) The computer-implemented system according to claim 4, wherein:  
the data structure is a queue.
7. (Currently Amended) The computer-implemented system according to claim 4, wherein:  
the data structure is processed using a first-in-first-out approach.
8. (Currently Amended) The computer-implemented system according to claim 1, wherein:  
the matching component ~~is operable to~~ keeps only a portion of the XML document in  
memory at any given time.
9. (Currently Amended) The computer-implemented system according to claim 1, wherein:  
the matching component ~~is operable to~~ knows the schema of the XML document and  
foreseeing the coming events.
10. (Currently Amended) The computer-implemented system according to claim 1, wherein:  
the match is an expression-based match, which can be an XPath query.
11. (Currently Amended) The computer-implemented system according to claim 3, wherein:  
the matching component ~~is operable to~~ keeps, clones and destroys the entirety or a  
portion of the sub-tree descending from a node in the tree.
12. (Currently Amended) The computer-implemented system according to claim 1, wherein:

the user object is ~~operable to~~ returns the matched event to an XML stream for use by any other component.

13. (Currently Amended) A method for processing an XML document, comprising:
- parsing an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;
  - accepting an event from the stream of events and keeping in memory only a ~~subset~~ said event of the stream of events at any one time;
  - performing a match on ~~an event in the subset~~ said event of the stream of events;
  - notifying an observer if when the event is a matched event, wherein when the event is not a matched event the observer is not notified;
  - listening for the matched event and passing it to a user object; and
  - handling the matched event.
14. (Previously presented) The method according to claim 13, further comprising:
- representing the XML document in a hierarchical structure, which is a tree with each node containing a portion of the document.
15. (Original) The method according to claim 14, wherein:
- the parsing of the XML document comprises the steps of:
    - traversing the XML tree and adding visited nodes into a data structure;
    - processing the nodes in the data structure and generating an event for each node;
    - and
    - appending the event to the output stream.
16. (Original) The method according to claim 15, wherein:
- the XML tree is traversed using a breath-first or depth-first search.
17. (Original) The method according to claim 15, wherein:
- the data structure is processed using a first-in-first-out approach.

18. (Original) The method according to claim 13, further comprising:  
keeping only a portion of the XML document in memory at any given time.
19. (Original) The method according to claim 13, further comprising:  
knowing the schema of the XML document and foreseeing the coming events.
20. (Previously presented) The method according to claim 13, further comprising:  
performing an expression-based match, which is an XPath query.
21. (Original) The method according to claim 14, further comprising:  
keeping, cloning and destroying the entirety or a portion of the sub-tree descending from a node in the tree.
22. (Previously presented) The method according to claim 13, further comprising:  
returning the matched event to an XML stream for use by any other component.
23. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:
- parse an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;  
accepting an event from the stream of events and keeping in memory only ~~a subset~~ said event of the stream of events at any one time;  
perform a match on ~~an event in the subset~~ said event of the stream of events;  
notify an observer if when the event is a matched event, wherein when the event is not a matched event the observer is not notified;  
listen for the matched event and pass it to a user object; and  
handle the matched event.

24. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:
- represent the XML document in a hierarchical structure, which can be a tree with each node containing a portion of the document.
25. (Original) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:
- parse the XML document, comprising the steps of:
    - traversing the XML tree and adding visited nodes into a data structure;
    - processing the nodes in the data structure and generating an event for each node;
    - and
    - appending the event to the output stream.
26. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:
- traverse the tree using a breath-first or depth-first search.
27. (Original) The machine readable medium of claim 25, further comprising instructions that when executed cause the system to:
- process the data structure using a first-in-first-out approach.
28. (Previously presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:
- perform an expression-based match, which is an XPath query.
29. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:
- keep only a portion of the XML document in memory at any given time.
30. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

know the schema of the XML document and foresee the coming events.

31. (Original) The machine readable medium of claim 24, further comprising instructions that when executed cause the system to:

keep, clone and destroy the entirety or a portion of the sub-tree descending from a node in the tree.

32. (Previously presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

return the matched event to an XML stream for use by any other component.

33. (Currently Amended) A computer-implemented system for processing an XML document, comprising:

means for parsing an XML document to generate a stream of events, wherein each event in the stream represents a portion of the document;

means for accepting an event from the stream of events and keeping in memory only a subset said event of the stream of events at any one time;

means for performing a match on ~~an event in the subset~~ said event of the stream of events;

means for notifying an observer if the event is a matched event, wherein when the event is not a matched event the observer is not notified;

means for listening for the matched event and passing it to a user object; and

means for handling the matched event.

34. (Canceled).

35. (New) The computer-implemented system according to claim 1, wherein:

said matching component can perform the step of accepting another event at said time.

36. (New) The computer-implemented system according to claim 1, wherein:

said matching component can perform the step of accepting another event at a different time.

37. (New) The method according to claim 13, further comprising:

accepting another event at said time.

38. (New) The method according to claim 13, further comprising:

accepting another event at a different time.

39. (New) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

accept another event at said time.

40. (New) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

accept another event at a different time.

41. (New) The computer-implemented system according to claim 33, further comprising:

means for accepting another event at said time.

42. (New) The computer-implemented system according to claim 33, further comprising:

means for accepting another event at a different time.